Visual Inertial SLAM

Sambaran Ghosal
Department of Electrical and Computer Engineering
UC San Diego
La Jolla, USA
sghosal@ucsd.edu

Nikolay Atanasov
Department of Electrical and Computer Engineering
UC San Diego
La Jolla, USA
natanasov@ucsd.edu

***Abstract*—In this project, we discuss the problem of Simultaneous Localisation and Mapping(SLAM) of a robot moving in an initially unknown environment. SLAM problem is a widely common thing in the field of robotics which enables us to track the robot location as well as simultaneously building a map of the environment that can be used for later purposes. There are many different approaches to solve the SLAM problem but we will be discussing the Visual Inertial SLAM problem in this project.**

***Index Terms*—SLAM, Bayes Filter, Kalman Filter, Extended Kalman Filter, Inertial Measurement Unit, Stereo Camera**

I. Introduction

A robot often has to move over uneven terrains such as valleys, hills, construction sites, congested labs etc. In all of these tasks we want to obtain a representation of the environment that the robot moves in that can be used for future purposes such as define some landmarks/goal points to instruct the robot to go to. But mapping cannot be done without knowing where the robot is at each instant. Sensor data provides us with observations only in the sensor frame and without knowing the position of the robot with respect to a global frame, these are not useful. Hence we can see that the Mapping requires knowing the trajectory of the robot at each time. Whereas to know the trajectory of the robot, we need to have the MAP. Hence we have in front of ourselves a chicken-egg problem.

In this project, we will see how to approach this problem using a Bayes Filter. Specifically, we will look at one of the implementation of the Bayes Filter which uses a gaussian distribtution as an estimate of the robot poses, landmarks of the map and using the motion model of the robot and observation model of the sensors mounted in the robot, update the distribution mean and covariance for both the robot pose as well as the landmarks present in the map. This is known as the **KALMAN FILTER**. A KalmanFilter can only be applied when the motion and observation model corresponding to the robot is linear with respect to the state. This is not a very good assumption to assume since real life systems are widely non-linear. Hence we apply an extension of the Kalman Filter called **EXTENDED KALMAN FILTER** that uses first order approximations to linearize the motion and the observation models.

II. Problem Formulation

The aim of this project is to implement an Extended Kalman Filter on a non-linear robot system to predict the trajectory of the robot as well as build a map of the surroundings of the robot. In this project, the robot is a car driving around the city streets, and the map of the surroundings is a collection of point cloud data that represent gradient/changing features in the surroundings. The car is equipped with an Inertial Measurement Unit(IMU) that gives us the linear and the angular velocity of the car in the body frame itself. It is also equipped with a stereo camera that is able to capture RGB images of the surroundings. Using these images the pixels of point cloud of changing features is provided to us directly as the observation model data.The robot pose and the position of the landmarks in the map are both assumed to be univariate gaussian distributions. The robot pose consists of the car position and the orientation with respect to the world frame. This is expressed as a $R^{4\times 4}$ matrix $\in$ SE(3) space. Hence the robot pose can be treated as a gaussian distribution with mean $\mu_{t|t}^{IMU} \in$ SE(3) space, and covariance $\Sigma_{t|t}^{IMU} \in R^{6\times 6}$. The landmarks can also be treated as a gaussian distribution. Suppose there are total M landmarks in the map that is seen over the horizon of the motion of the robot. Each landmark is described the the three coordinates of its position in the world frame. Hence the full map of the environment can be treated as a gaussian pdf with mean $\mu_{t|t}^{MAP} \in R^{3\times M}$, and covariance $\Sigma_{t|t}^{MAP} \in R^{3M\times 3M}$.

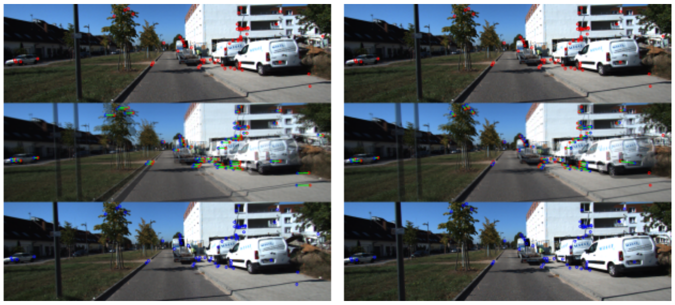
Fig. 1: Visual features as seen from the stereo camera of the car

Given a sequence of control inputs $u_{0:t-1}$ and observation $z_{1:t}$), obtain the trajectory of the robot $p(x_t|z_{1:t}, u_{0:t-1})$ and the map of the robot $p(m|x_t, z_{1:t}, u_{0:t-1}$. Using the Markov Factorization, we can write the joint distribution of the map and the trajectory as

$$p(m, x_{0:t}, z_{1:t}, u_{0:t-1}) = p_{0|0}(x_0, m) \prod_{i=1}^{t} p_h(z_i|x_i) \prod_{i=0}^{t-1} p_f(x_{i+1}|x_i, u_i) \quad (1)$$

where $p_h(z_t|x_t, m)$ is the observation model of the robot, and $p_f(x_t|x_{t-1}, u_{0:t-1})$.

Then the SLAM problem can be formulated as a Maximum Likelihood problem as follows to obtain the map and the trajectory of the robot

$$\max_{x_{0:T,m}} \sum_{t=0}^{T} log\, p_h(z_t|x_t, m) + \sum_{t=1}^{T} log\, p_f(x_t|x_{t-1}, u_{0:t-1}) \quad (2)$$

III. Technical Approach

There are three major steps involved in the Visual Inertial SLAM process :-

- First, we have to predict the IMU mean pose $\mu^{IMU}_{t+1|t}$ and covariance matrix of the pose $\Sigma^{IMU}_{t+1|t}$ given the IMU measurement at time t using the Extended Kalman Filter predict equations. Additionally we assume some noise in the IMU measurements. This noise contributes to the update of the covariance of the IMU pose during the predict step.
- Next we obtain the landmark position mean $\mu^{MAP}_{t+1|t+1}$ and covariance matrix $\Sigma^{MAP}_{t+1|t+1}$ by applying the update step equations of the Extended Kalman Filter using the stereo camera equation of the left and right image as the observation model.
- Using the stereo camera input at time $t+1$, we update the estimated pose and covariance during the predict step to obtain the corrected pose $\mu^{IMU}_{t+1|t+1}$ and corrected covariance matrix $\Sigma^{IMU}_{t+1|t+1}$. The same stereo camera equation model is used as the observation model but this time the differentiation is computed with respect to the pose of the IMU.

We will first describe how to do the predict step of the EKF to obtain the predicted pose and covariance of the IMU, then see how to update the landmarks of the map and then update the pose and covariance of the IMU using the stereo camera observations independently of the landmark update step. Finally we will describe the method to do SLAM where we simultaneously update the landmarks as well as the IMU pose and covariance. This is the Visual SLAM part of the project. Let us first describe how to do the three parts of the problem independently :-

A. IMU Pose Prediction

Assuming that the prior pose of the IMU is known as a gaussian distribution with mean $\mu^{IMU}_{t|t}$, covariance $\Sigma^{IMU}_{t|t}$ i.e. ${}_wT_{imu} \sim N(\mu^{IMU}_{t|t}, \Sigma^{IMU}_{t|t})$, given a new IMU measurement of linear and angular velocities at time $t+1$, the estimated robot pose after moving is given by the EKF predict step equations. Assuming a white noise of covariance W in the control input, the predict step equations can be written as follows

$$\mu^{IMU}_{t+1|t} = \mu^{IMU}_{t|t} e^{\tau \hat{u_t}} \quad (3)$$

where $u_t = [v_t, \omega_t]^T$ is the control input at time t, $\hat{u_t}$ is defined as the 4×4 twist matrix defined as

$$\hat{u_t} = \begin{bmatrix} \hat{\omega_t} & v_t \\ 0 & 0 \end{bmatrix} \quad (4)$$

where again, $\hat{}$ is defined as the hat map operator that takes a vector from R^3 space to the $so(3)$ space of skew-symmetric matrices as follows

$$\hat{x} = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \quad (5)$$

The noise in the IMU measurement is treated separately using perturbation kinematics which contributes the covariance matrix of the predict step. The predicted covariance matrix is given as follows

$$\Sigma^{IMU}_{t+1|t} = (e^{-\tau \overset{\curlywedge}{u_t}}) \Sigma^{IMU}_{t|t} (e^{-\tau \overset{\curlywedge}{u_t}})^T + W \quad (6)$$

where $\overset{\curlywedge}{u_t}$ is defined as

$$\overset{\curlywedge}{u_t} = \begin{bmatrix} \hat{\omega_t} & \hat{v_t} \\ 0 & \hat{\omega_t} \end{bmatrix} \quad (7)$$

and W is the noise covariance matrix associated with the motion model, i.e. the noise in IMU measurements which is $R^{6\times6}$ matrix, defining noise in the linear and angular velocities.

Thus we have completed the predict step of the EKF on the motion model of the pose of the IMU. Next, let us discuss the update step of the EKF to get the landmark positions in the map from the stereo camera visible features at a given time.

B. Landmark Positions Update

Our robot is equipped with a stereo camera that captures images of the environment the robot moves in. From these images we are given a collection of points where there is a change in feature such as color intensity. Using these point clouds as our stereo camera observations, our goal is now to obtain the world frame coordinates of these point clouds. We assume there is some noise in the pixel values obtained from the stereo camera. Let the noise be a white noise with zero mean and covariance V in each pixel dimension. At a given time, out of the M total landmarks over the total horizon of time of motion, assuming only N_t features are visible, corresponding to each of these visible features, we can write the stereo camera equation relating the landmark world coordinates to the pixel coordinates as

$$z_{t+1,i} = K_s \pi({}_{cam}T_{imu\;imu}T_w m_i) + N(0, I_{4,4} \times V) \quad (8)$$

where $z_{t+1,i}$ is the stereo camera pixels of the i^{th} visible feature at time $t+1$, which contains the left and right stereo camera pixels, K_s is the stereo camera matrix built from the camera intrinsic calibration matrix, π is the projection function that takes a 4 dimenional vector and divides by the third component of the vector, and $I_{4,4}$ is the 4×4 identity matrix to add the noise term in each of the pixel dimension. Given that the predicted IMU pose is given as $\mu_{t+1|t}^{IMU}$, and the landmark position m_i, we can update the estimate of the landmark position as follows using the EKF update steps

$$\tilde{z}_{t+1,i} = K_s \pi({}_{cam}T_{imu}\, \mu_{t+1|t}^{IMU-1}\, m_i) \tag{9}$$

where $\tilde{z}_{t+1,i}$ is the expected feature value assuming the predicted IMU pose and previous estimate of landmark position. Because of the noise in the stereo camera, there is a difference between the expected feature values and the given feature values. This difference is called the innovation.

$$r_{t+1,i} = z_{t+1,i} - \tilde{z}_{t+1,i} \tag{10}$$

We define the Kalman Gain for the update step of the EKF as follows for the whole set of visible observation as follows.

$$K_{t+1|t} = \Sigma_{t|t}^{MAP} H_{t+1}^T (H_{t+1} \Sigma_{t|t}^{MAP} H_{t+1}^T + I_{4N_t,4N_t} \times V)^{-1} \tag{11}$$

The Kalman Gain defines how much we should change the estimate of our interested variable based on the noise of the observed quantity. If the noise is less, the sensor is deemed trustworthy and hence the Kalman gain is large and it will change the value of the interested variable significantly. On the other hand if the sensor is noisy, the sensor is deemed not so trusty and hence there is insignificant change in the interested variable. Using the kalman gain, the mean and covariance matrix of the visible landmark features are updated as follows

$$\mu_{t+1|t+1}^{MAP} = \mu_{t|t}^{MAP} + K_{t+1|t} r_{t+1} \tag{12}$$

$$\Sigma_{t+1|t+1}^{MAP} = (I - K_{t+1|t} H_{t+1}) \Sigma_{t|t}^{MAP} \tag{13}$$

In the above equations, H_{t+1} is the jacobian of the predicted observation with respect to the landmark positions m_i for $i \in \{1, 2, ..., N_t\}$ defined as follows

$$H_{t+1,i} = K_s \frac{d\pi(q)}{dq}\, {}_{cam}T_{imu}\, \mu_{t+1|t}^{IMU-1}\, P^T \tag{14}$$

where q is the optical frame coordinates of m_i and P is the matrix defined as $\begin{bmatrix} I_{3\times3} \\ 0 \end{bmatrix}$. We have N_t visible features and hence the total jacobian will be a matrix $\in R^{4N_t \times 3M}$.

Hence, using the equations (9) to (14) and the given features, we can update the landmark positions and the covariance associated with these landmarks. This concludes our update step for landmarks estimation. Now let us discuss the aspect of updating the IMU pose using the features visible from the stereo camera.

C. IMU Pose Update

After we have a predicted pose for the IMU from the EKF Predict step using the IMU measurements (Eq(3) and Eq(6), we can correct the IMU pose again using the stereo camera observations that we get at time $t + 1$. For this we assume that the landmark positions are known to us. The observation model remains the same as that of the landmark update step, except now we will be computing the Jacobian of the observation model with respect to the mean pose of the IMU. We then compute the kalman gain for the pose update step and use it to compute the new mean pose $\mu_{t+1|t+1}^{IMU}$ and new covariance of the pose $\Sigma_{t+1|t+1}^{IMU}$. The steps are as follows

$$\tilde{z}_{t+1,i} = K_s \pi({}_{cam}T_{imu}\, \mu_{t+1|t}^{IMU-1}\, m_i) \tag{15}$$

$$H_{t+1,i} = -K_s \frac{d\pi(q)}{dq}\, {}_{cam}T_{imu} (\mu_{t+1|t}^{IMU-1} m_i)^{\circ} \tag{16}$$

where for $s \in R^{4\times1}$ vector, the $\circ$ operator is defined as a mapping to $R^{6\times6}$ as shown below

$$s^{\circ} = \begin{bmatrix} I_{3,3} & -\hat{s} \\ 0 & 0 \end{bmatrix} \tag{17}$$

Again assuming we have N_t visible features, the jacobian of the observation model with respect to the pose of the IMU will be a $R^{4N_t \times 6}$ matrix. Using the jacobian, now we can do the update steps for the IMU pose as follows

$$K_{t+1|t} = \Sigma_{t+1|t}^{IMU} H_{t+1}^T (H_{t+1} \Sigma_{t+1|t}^{IMU} H_{t+1}^T + I_{4N_t,4N_t} \times V)^{-1} \tag{18}$$

$$\mu_{t+1|t+1}^{IMU} = \mu_{t+1|t}^{IMU} e^{(K_{t+1|t} r_{t+1})^{\hat{}}} \tag{19}$$

$$\Sigma_{t+_1|t+1}^{IMU} = (I - K_{t+1|t} H_{t+1}) \Sigma_{t+1|t}^{IMU} \tag{20}$$

This way we can have now a full approach to use the EKF filter to do first predict the IMU pose after the car moves, then using the stereo camera images, correct the IMU pose as well as correct the previous estimates of the landmark poses. The one problem associated with doing the update step separately is that for IMU pose update, we have to assume that the landmark pose is known to us correctly, and for the landmark update we have to assume that the IMU pose is known to us correctly. In reality, both these are known only as a probabilistic estimate and there is an error associated with this. Hence the update step following this procedure will lead to errors. Hence we do the Simultaneous Localisation and Mapping(SLAM) where we simultaneously update both the landmark positions as well as the IMU pose. Let us see the detailed steps of the procedures to do SLAM which in our project is called Visual SLAM because we are using visual features as our observation model.

D. Visual Inertial SLAM

In Visual Inertial SLAM, the predict step still remains the same as we have discussed is section III(A). We use the IMU measurements to predict where the car may be located at time $t + 1$ given a prior of where the car was located at time t. The only difference is that instead of updating the landmark positions and IMU pose separately, we try to update them at once. The approach is to create a joint probability distribution of the IMU pose and the landmark position as one full big state for the EKF. Hence we construct a covariance matrix that is $R^{(3M+6)\times(3M+6)}$ contains the $R^{3M\times3M}$ map covariance matrix along one of the block diagonals, and the $R^{6\times6}$ IMU pose covariance in the last diagonal block. In addition to the map and IMU covariance, we also model the correlation between the map features and the IMU pose. This is captured in the off digaonal block matrices in the big covariance matrix. Hence the structure of our new covariance matrix is as follows

$$\Sigma_{SLAM} = \begin{bmatrix} \Sigma_{MAP} & \Sigma_{MAP,IMU} \\ \Sigma_{IMU,MAP} & \Sigma_{IMU} \end{bmatrix} \tag{21}$$

We now need to get the combined Jacobian for both the MAP and IMU pose to do the SLAM. This process is just concatenating the jacobians we found in Eq(14) and Eq(16) along the columns.

$$H_{SLAM} = \begin{bmatrix} H_{MAP} & H_{IMU} \end{bmatrix} \tag{22}$$

We know that $H_{MAP} \in R^{4N_t \times 3M}$ and $H_{IMU} \in R^{4N_t \times 6}$ matrices, hence the $H_{SLAM} \in R^{4N_t \times (3M+6)}$ matrix.

Now we have the jacobian and the covariance matrix of the map and IMU combined, we can again apply the EKF predict and update step as follows.

1) SLAM predict step:

$$\mu^{IMU}_{t+1|t} = \mu^{IMU}_{t|t} e^{\tau \hat{u}_t} \tag{23}$$

$$\Sigma_{MAP} = \Sigma_{MAP} \tag{24}$$

$$\Sigma_{IMU} = (e^{-\tau \overset{\curlywedge}{u}_t}) \Sigma_{IMU} (e^{-\tau \overset{\curlywedge}{u}_t})^T + W \tag{25}$$

$$\Sigma_{MAP,IMU} = \Sigma_{MAP,IMU} \, (e^{-\tau \overset{\curlywedge}{u}_t})^T \tag{26}$$

2) SLAM update step:

$$S = H_{SLAM} \Sigma_{SLAM} H^T_{SLAM} + I_{4N_t,4N_t} V \tag{27}$$

$$K_{SLAM} = \Sigma_{SLAM} H^T_{SLAM} S^{-1} \tag{28}$$

The mean of the IMU pose and landmark positions can now be updated the same way as done in Eq(19) and Eq(12) except that now we will use the corresponding blocks of Kalman gain from Eq(28) which has also modelled the correlations between the map features and the IMU pose and hence is a better estimator of the state values.

IV. Results

We are given two trajectory files for the car, 10.npz and 03.npz that represent the car moving in two different places. We first present the results for independent update of the IMU and the landmarks for each file. Later we present the results of Visual Inertial SLAM on both the files.

A. Independent Update of IMU pose and landmark positions

1) 10.npz: In this section, we are going to see the effect of EKF predicition step to get the IMU pose using the given IMU data for the car, and then use this to update the landmark positions of the map for the 10.npz dataset provided to us.

2) 03.npz: Same approach is now done for the 03.npz dataset.

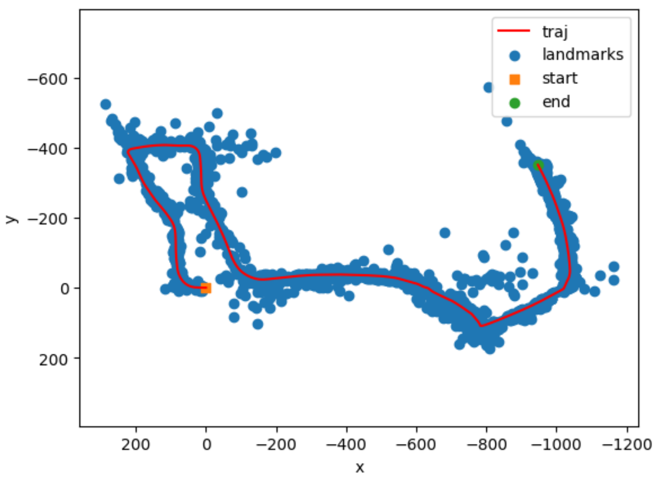

Fig. 2: IMU Pose using EKF Predict step for 10.npz

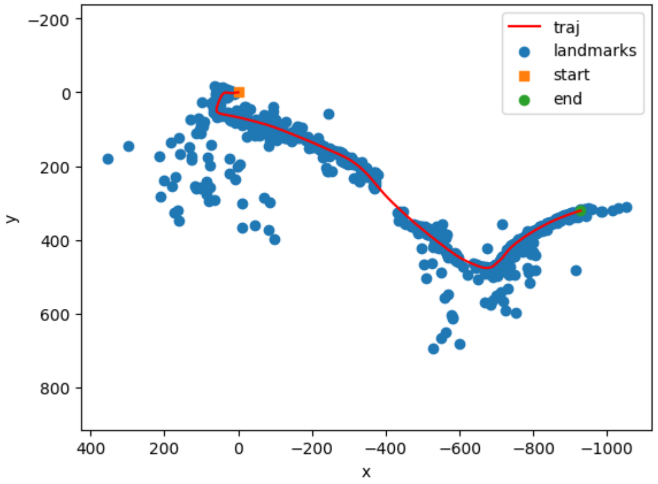

Fig. 3: IMU Pose using EKF Predict step for 03.npz

B. Visual SLAM for IMU pose and landmark positions

Now we will see the results for the combined update of the IMU pose as well as the landmarks using the IMU data and the stereo image data. The trajectories are expected to change a lot but their behaviour should still remain the same (for ex : direction of turns).

1) 10.npz: The combined update of IMU pose and landmark is shown below. We can see the trajectory changes significantly but the overall behaviour still remains the same. The IMU data is typically not very accurate and also suffers from the problem of drift. Hence having the stereo camera image to update the pose has a large correction factor that places the trajectory in the correct position according to the innovation of the observation model.

2) 03.npz: Same procedure for SLAM was applied to the 03.npz dataset and the trajectory obtained is as shown below

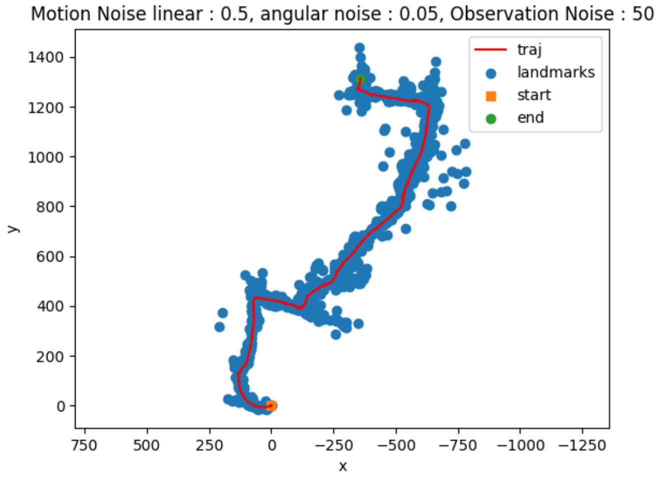

Fig. 4: Visual SLAM for 10.npz

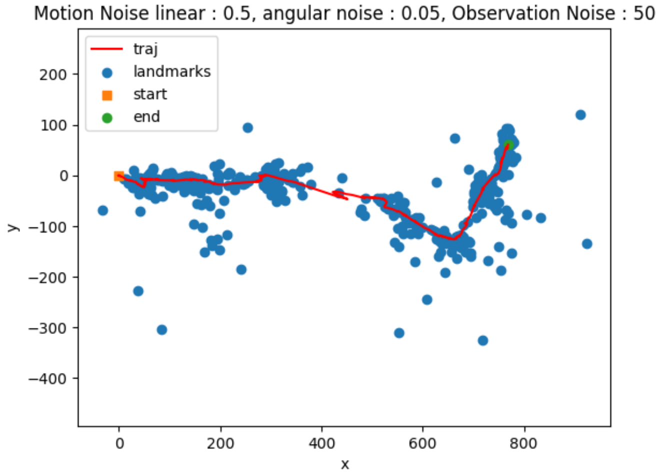

Fig. 5: Visual SLAM for 03.npz

C. *Effect of motion noise on trajectory*

For the Kalman Filter, the motion noise plays an important role in determining the covariance of the IMU pose. This later affects the kalman gain matrix that is used to update the mean as well as the covariance matrix. A too low value of noise may result in a very high Kalman Gain and hence we put very large trust in our sensor even though the sensor may actually be quite noisy. On the other hand, if the kalman gain is too small, we dont change the values too much and hence put very less trust on the sensor even though it may be quite accurate. Hence we need to tune the motion covariance matrix carefully. Fig(6) shows some experimental results of the effect of motion noise W on both the 10.npz data and 03.npz data.

D. *Effect of observation noise on trajectory*

Another important parameter affecting the Kalman Filter equations is the observation noise V. Here our noise lies in the pixels space. Depending on the amount of observation noise, the kalman gain can either be small or large. If our observation sensor is trustworthy, we would want to assign low noise to the observation model and hence the Kalman gain will be able to correct the poses significantly. However if the sensor is noisy, we would assign large value to the observation noise and this would make the Kalman gain small and we would not assign super importance to the observations. Making the noise too small or too large on the other hand is not a good idea because we don't exactly know the exact behaviour of the sensor. Hence a fine tuned value for the observation noise is important for the SLAM process to work out. Results for several choices of V for both 03.npz and 10.npz are shown in Fig(7).

V. Conclusion and Future Work

In this project, we implemented a visual inertial SLAM method to keep track of the pose of the car as well as create a map of the environment that is the point cloud of a set of feature points. We particularly used Extended Kalman Filter equations to do the predict and update steps because the system is non-linear. We used rotation and pose kinematics and geometry i.e. SE(3) and SO(3) kinematics and SE(3) perturbations.

For future work, we could look at other versions of a Kalman Filter such as the Unscented Kalman Filter(UKF). We could also try to implement something like a Mixture of Kalman Filters so that we could create multiple hypothesis for the robot and landmark poses and then select the best ones. Another important factor that could be accounted for in future study is the **Gaiting Test** that can be used to decide which features to actually consider at a given time based on its distance from the robot.

VI. ACKNOWLEDGEMENT

I would like to thank Professor Nikolay Atanasov for his help and suggestions throughout this project. His assistance helped me overcome some crucial problems regarding some key mathematical concepts as well as implementation details encountered during the project. I would also like to acknowledge Mr. Samveed Desai and Mr Alexander Toofanian with whom I had lots of whiteboard discussions, brainstorming sessions and discussions about some key aspects of the implementation of the project and data visualisation.

References

[1] Nikolay Atanasov, "https://natanaso.github.io/ece276a/ref/ECE276A_7_Rotations.pdf"
[2] "https://natanaso.github.io/ece276a/ref/ECE276A_8_MotionAndObservationModels.pdf
[3] "https://natanaso.github.io/ece276a/ref/ECE276A_9_BayesianFiltering.pdf"
[4] "https://natanaso.github.io/ece276a/ref/ECE276A_12_EKF_UKF.pdf"
[5] "https://natanaso.github.io/ece276a/ref/ECE276A_14_SO3_SE3.pdf"
[6] "https://natanaso.github.io/ece276a/ref/ECE276A_13_VI_SLAM.pdf"

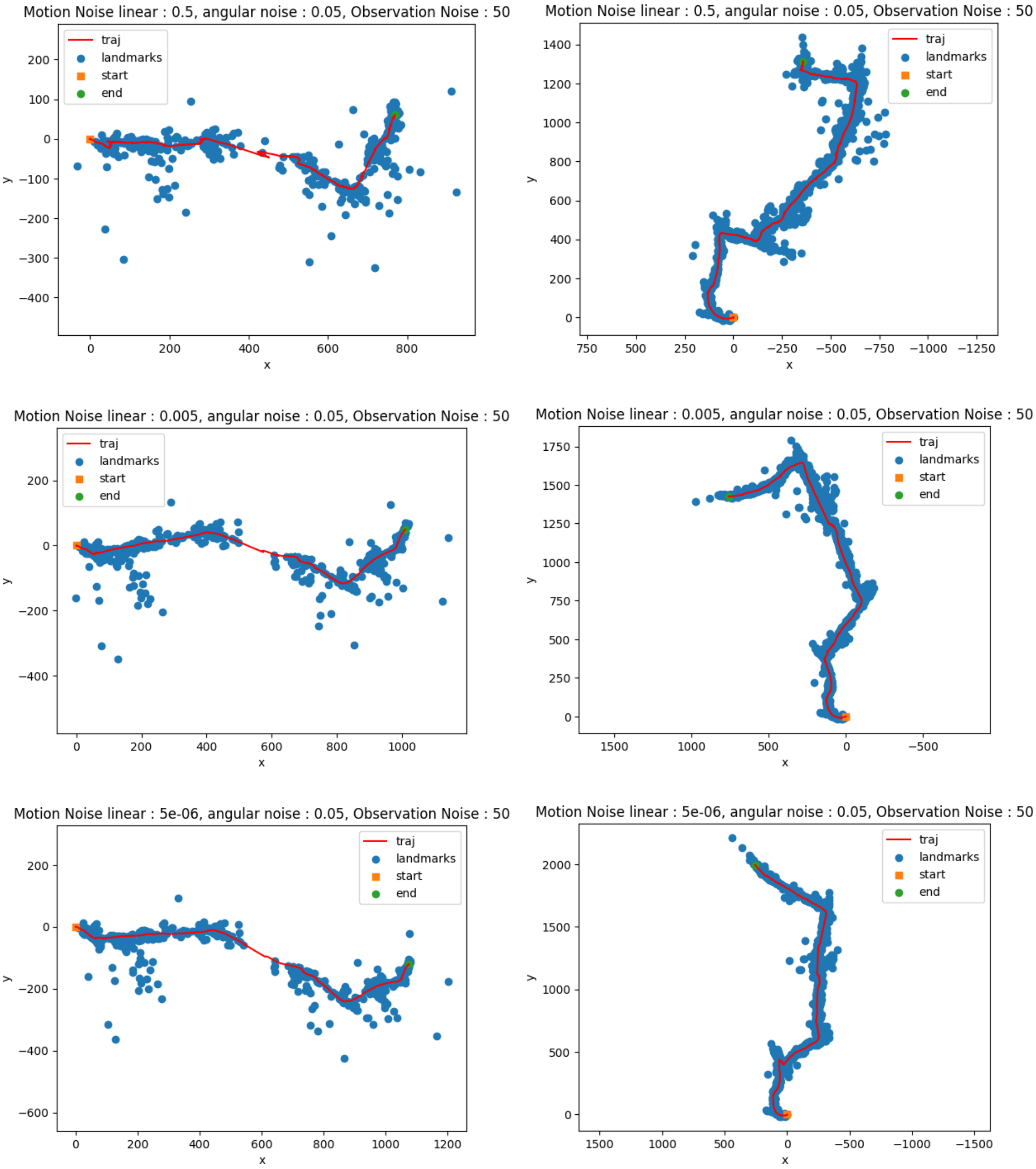

Fig. 6: Effect of motion noise in SLAM, 03.npz left and 10.npz right

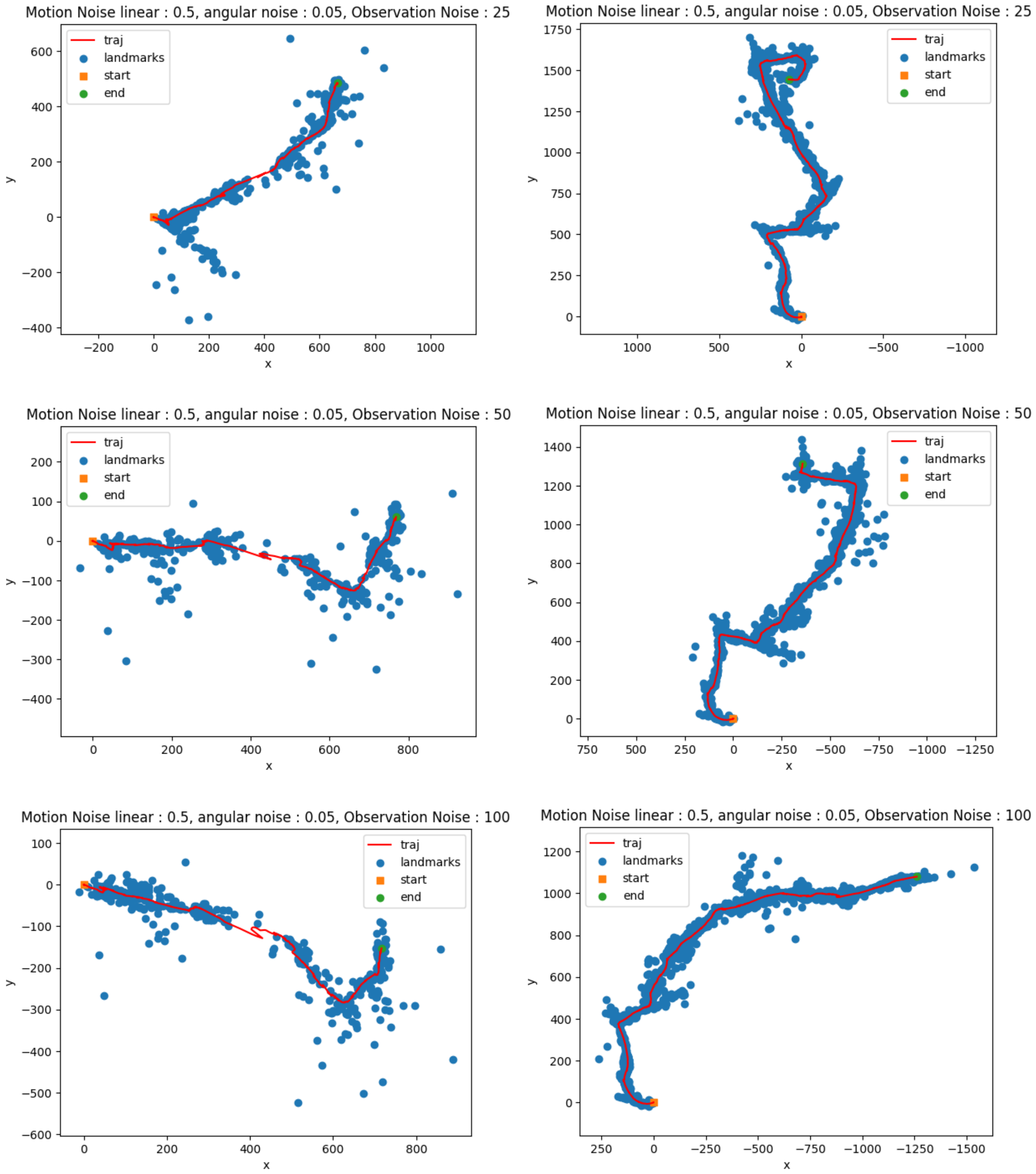

Fig. 7: Effect of observation noise in SLAM, 03.npz left and 10.npz right